SELECT package demonstration  
Covered codend

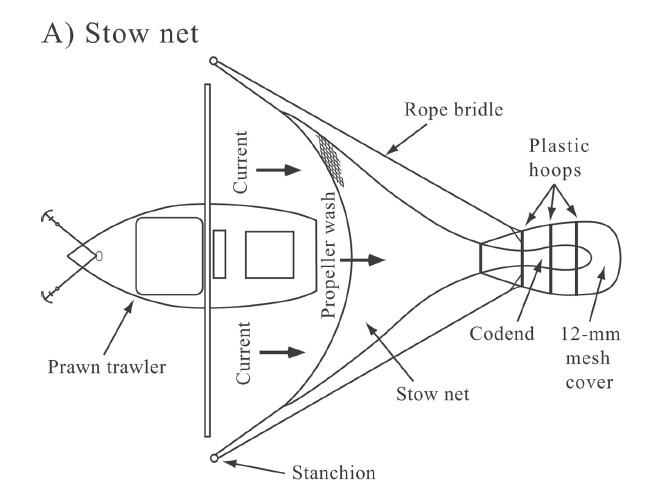
## Summary

**This case study demonstrates:**

* Fitting of logistic and Richards selection curves to covered-codend data.
* Handling sub-sampled data.
* Using combined individual-haul data to estimate over-dispersion (due to causes such as sub-sampling, between-haul variability and non-independent behaviour of fish).
* Using the bootstrap and over-dispersion adjustments.
* BONUS section: Fitting a semi-parametric spline selection curve.

### Data source

The data are for school prawn selectivity in a 30 mm square-mesh stow-net trawl (MacBeth et al. 2005. Improving selectivity in an Australian penaeid stow-net fishery. Bulletin of Marine Science, 76: 647-660).



### Load required packages

require(SELECT)  
require(dplyr)

### Input the data

#Read in data and remove zero catch lengths  
Stow.df=read.csv("../Data/LgSq.csv",header=T) #343 rows  
Stow.df=subset(Stow.df,total>0) #213 rows  
  
#Quick peek at data  
head(Stow.df)

## lgth codend cover WtSch qcodend qcover total Haul  
## 6 11 0 2 2.02 0.585 0.34 2 1  
## 8 13 2 8 2.02 0.585 0.34 10 1  
## 9 14 3 6 2.02 0.585 0.34 9 1  
## 10 15 25 34 2.02 0.585 0.34 59 1  
## 11 16 47 82 2.02 0.585 0.34 129 1  
## 12 17 65 62 2.02 0.585 0.34 127 1

Note the sampling fractions qcodend and qcover - the catch was sub-sampled.

### Define variable names, and the names of sampling fraction variables

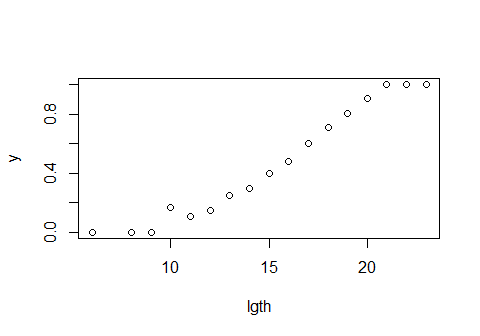
v.names=c("lgth","cover","codend")  
q.names=c("qcover","qcodend")

### Produce a plot of retention proportions

Tots=Raw2Tots(Stow.df,v.names,q.names)  
Tots=transform(Tots,n=codend+cover,y=codend/(codend+cover))   
Tots

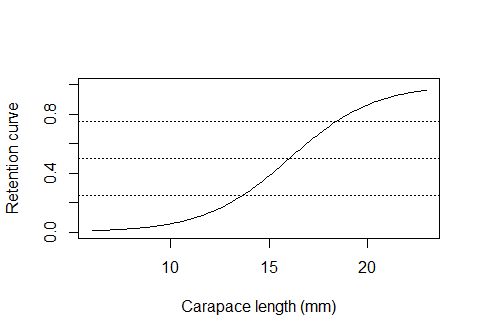
## lgth cover codend n y  
## 1 6 2.239274 0.000000 2.239274 0.0000000  
## 2 8 1.502768 0.000000 1.502768 0.0000000  
## 3 9 5.259689 0.000000 5.259689 0.0000000  
## 4 10 7.513842 1.502768 9.016610 0.1666667  
## 5 11 35.978043 4.389272 40.367315 0.1087333  
## 6 12 42.402885 7.235204 49.638089 0.1457591  
## 7 13 71.259699 23.271511 94.531210 0.2461781  
## 8 14 131.173924 55.648326 186.822250 0.2978678  
## 9 15 359.150971 235.007945 594.158916 0.3955305  
## 10 16 729.029720 679.320778 1408.350498 0.4823521  
## 11 17 619.185516 921.883354 1541.068870 0.5982104  
## 12 18 338.090283 819.113029 1157.203312 0.7078385  
## 13 19 102.530989 426.214007 528.744996 0.8060861  
## 14 20 14.358586 140.110820 154.469406 0.9070458  
## 15 21 0.000000 36.272249 36.272249 1.0000000  
## 16 22 0.000000 11.034010 11.034010 1.0000000  
## 17 23 0.000000 1.320535 1.320535 1.0000000

plot(y~lgth,data=Tots) #las=1 gives horizontal y tick values

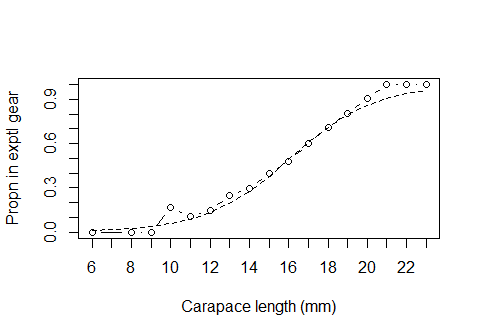
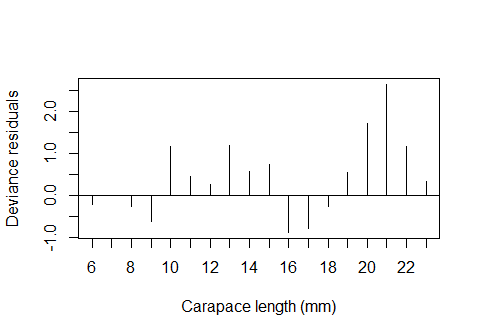


### Fit a logistic selection curve, plot it, and do model checks

Logist.fit=SELECT(Stow.df,var.names=v.names,q.names=q.names)  
plot(Logist.fit, xlab="Carapace length (mm)")



ModelCheck(Logist.fit, xlab="Carapace length (mm)")



## Model fit:  
## null.l model.l full.l npar AIC   
## -370.68605 -38.60693 -29.77578 2.00000 81.21385   
## GOF:  
## Deviance Pearson.chisq dof Deviance.CF Pearson.CF   
## 17.6623038 13.8283885 15.0000000 1.1774869 0.9218926

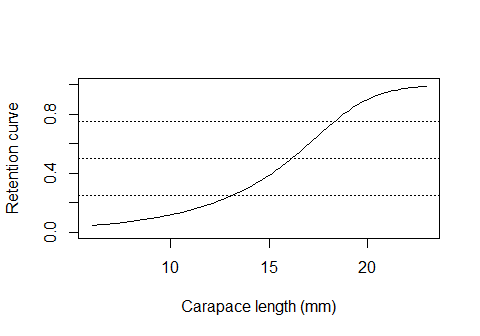
Estimates(Logist.fit)

## par raw s.e.  
## L50 16.051802 0.06689251  
## SR 4.746722 0.20555085

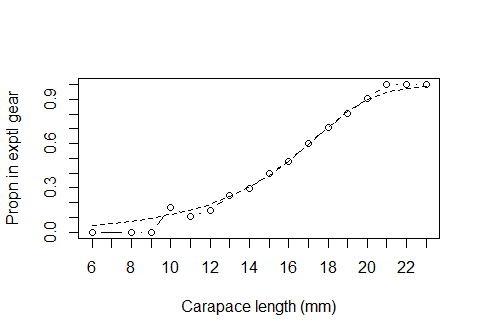
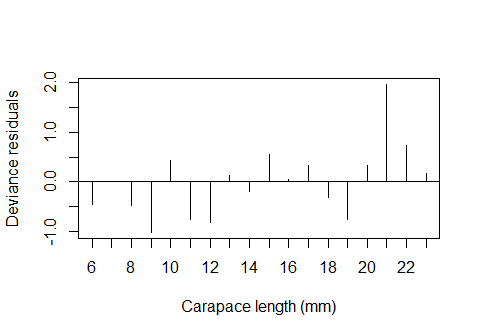
The logistic fit estimates L50 and SR of 16.1 and 4.7 mm, respectively. However, the deviance residual plot does suggest some mild lack of fit due to blocks of positive residuals. So, let’s try an asymmetric Richards curve fit.

### Fit a Richards selection curve

Rich.fit=SELECT(Stow.df,var.names=v.names,q.names=q.names,stype="richards")  
plot(Rich.fit, xlab="Carapace length (mm)")



ModelCheck(Rich.fit, xlab="Carapace length (mm)")



## Model fit:  
## null.l model.l full.l npar AIC   
## -370.68605 -34.03063 -29.77578 3.00000 74.06125   
## GOF:  
## Deviance Pearson.chisq dof Deviance.CF Pearson.CF   
## 8.5097014 5.6168274 14.0000000 0.6078358 0.4012020

Estimates(Rich.fit)

## par raw s.e.  
## L25 13.165287 0.31172347  
## L50 16.170395 0.07621766  
## L75 18.335320 0.07830442  
## SR 5.170033 0.33887230  
## delta 3.318629 1.19486367

The Richards fit eliminates the pattern in the residuals. The Richards fit estimates L50 and SR of 16.2 and 5.2 mm, respectively. Note the asymmetry in L25 and L75 around L50.

## Correcting for over-dispersion

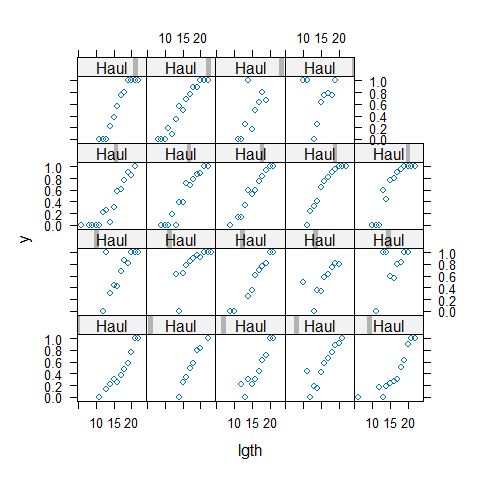
**CAUTION:** The log-likelihood and AIC from these fits to the summed data can **not** be used for inference because over-dispersion due to between-haul variation is likely and has not been taken into account. **BHV can not be detected from the summed data**

### Produce a lattic plot of retention proportions by haul

HaulCatch=Raw2Tots(Stow.df,v.names,q.names,sumHauls=F)  
HaulCatch=transform(HaulCatch,n=codend+cover,y=codend/(codend+cover))   
require(lattice) #For xyplot

## Loading required package: lattice

xyplot(y~lgth | Haul,data=HaulCatch)



Indeed, considerable between-haul (BHV) variation is present.

### Fit a Richards selection curve to the combined individual-haul data

Fitting to the combined individual-hauls data allows for proper detection of between-haul variability.

Rich.fit2=SELECT(Stow.df,var.names=v.names,q.names=q.names,stype="richards",  
 sumHauls=F)  
ModelCheck(Rich.fit2, xlab="Carapace length (mm)", minE=5, plots=F)

## Model fit:  
## null.l model.l full.l npar AIC   
## -984.2317 -647.5763 -218.2059 3.0000 1301.1526   
## GOF:  
## Deviance Pearson.chisq dof Deviance.CF Pearson.CF   
## 858.740853 814.138287 210.000000 4.089242 3.876849   
##   
## Correction factors from cells with expected count > 5 :  
## Deviance Pearson.chisq dof Deviance.CF Pearson.CF   
## 742.15150 714.76893 71.00000 10.45284 10.06717

The variance correction factor Pearson.CF shows that the actual variabilty could about 10 times that assumed under the binomial model.

With this amount of variability a bootstrap should be reasonable for appropriately quantifying the variability in the estimated parameters. (The bootstrap over-estimates parameter uncertainty if BHV is slight.)

### Define function to return quantities of interest: L25, L50, L75 and SR

ParFunc=function(data,var.names,q.names){  
 Rich.fit=SELECT(data,var.names,q.names=q.names,stype="richards")  
 Estimates(Rich.fit)[1:4,"par"]  
}  
#Check that it works  
ParFunc(Stow.df,v.names,q.names)

## L25 L50 L75 SR   
## 13.165287 16.170395 18.335320 5.170033

### Implement a bootstrap

nsim=99 #Use at least 999 in practice  
BootPars=bootSELECT(Stow.df,v.names,q.names,ParFunc,haul="Haul",paired=T,  
 nsim=nsim,verbose=F) #Use verbose=T to see progress

## Warning in sqrt(diag(varpars)): NaNs produced

##   
## Bootstrap successfully completed

cat("Bootstrap estimated standard errors are",apply(BootPars,2,sd))

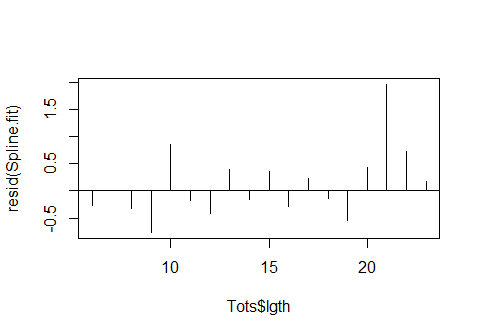
## Bootstrap estimated standard errors are 0.4738782 0.4897114 0.4879834 0.5491962

## Bonus analysis

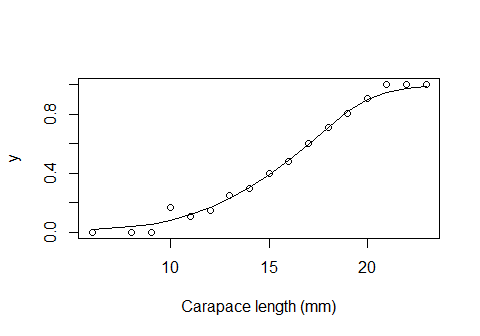
The Richards fit does estimate somewhat high retention for very small prawns, so here we explore whether the performance of a semiparametric fit using the SELECT function SplineSELECT.

### Fit a spline selectivity curve

require(mgcv)  
Spline.fit=SplineSELECT(Stow.df,var.names=v.names,q.names=q.names,  
 k=5,quasi=F)  
plot(Tots$lgth,resid(Spline.fit),type="h")  
abline(h=0)



plot(y~lgth,data=Tots,xlab="Carapace length (mm)")  
points(Tots$lgth,fitted(Spline.fit),type="l")



AIC(Spline.fit)

## [1] 75.24955

The spline fit achieves a nice fit, but does have somewhat higher AIC than the Richards. I